A NEW ALGORITHM FOR CONSTRAINED FINITE TIME OPTIMAL CONTROL OF HYBRID SYSTEMS WITH A LINEAR PERFORMANCE INDEX

Mato Baotić, Frank J. Christophersen, Manfred Morari

Automatic Control Laboratory, ETH Zentrum, ETL K 12, CH – 8092 Zürich, Switzerland
baotic | fjc | morari @control.ee.ethz.ch

Keywords: constrained finite time optimal control, discrete time systems, linear hybrid systems, dynamic programming, multi-parametric linear program.

Abstract

In this paper we present a modification of the algorithm described in [1, 2] for computing the solution to the constrained finite time optimal control problem for discrete time linear hybrid systems. As opposed to the quadratic performance index used in the original algorithm here we use a linear performance index. The algorithm combines a dynamic programming strategy with a multi-parametric linear program solver. By comparison with literature results it is shown that the algorithm presented here solves the considered class of problems in a computationally efficient way.

1 Introduction

In the last few years several different techniques were developed for the analysis and controller synthesis for hybrid systems [3, 4, 5, 6, 2]. A significant amount of the research in this field has focused on solving constrained optimal control problems, both for continuous time and discrete time hybrid systems.

We consider the class of discrete time linear hybrid systems, in particular, the class of constrained *piecewise affine systems* (PWA) that are obtained by partitioning the state space into polyhedral regions and associating to each region a different affine state update equation, cf. [4]. For such a class of systems the *constrained finite time optimal control* (CFTOC) problem can be solved by means of multi-parametric programming [2]. The solution is a piecewise affine state feedback control law and can be computed by using *multi-parametric mixed-integer quadratic programming* (mp-MIQP) for a quadratic performance index and *multi-parametric mixed-integer linear programming* (mp-MILP) for a linear performance index.

A novel computationally efficient method to solve the constrained finite time optimal control problem with the performance index based on the squared 2-norm was recently proposed by Borrelli *et al.* [1]. There the authors solved the problem backwards in time with the dynamic programming strategy where at each time step a multi-parametric quadratic program (mp-QP) is solved.

In this paper we present a modification of the aforementioned method [1] for optimal control problems with performance indices based on the 1- and ∞-norm. The algorithm uses a dynamic programming strategy together with a multi-parametric linear program solver (mp-LP).

The algorithm is compared with the mp-MILP solver proposed by Dua and Pistikopoulos [7]. It is impossible to conclude on the general superiority of one particular method. However, extensive simulations have shown that the presented algorithm is faster, less sensitive to ill-conditioned problems, and gives a more compact representation of the solution.

The aim of this paper is to give insight into the new approach and explain why it performs better for the class of problems we are considering.

2 Finite Time Constrained Optimal Control of linear Hybrid Systems

Consider the class of linear discrete time hybrid systems which can be stated as constrained piecewise affine systems of the following form

$$x(t+1) = f_{\text{PWA}}(x(t), u(t)) = A_i x(t) + B_i u(t) + f_i, \quad \text{if} \quad \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} \in \mathcal{P}_i := \{ \left[\begin{smallmatrix} x \\ u \end{smallmatrix}\right] \mid H_i x + J_i u \le K_i \} \tag{1}$$

where $t \geq 0$, $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the control input, and $\{\mathcal{P}_i\}_{i=1}^s$ is the polyhedral partition of the sets of the extended state+input space $\mathbb{R}^{n+m}$. Furthermore let the union of the polyhedral partitions be $\mathcal{P} := \cup_{i=1}^s \mathcal{P}_i$. Note that linear state and input constraints of the general form $Kx(t) + Lu(t) \leq M$ are incorporated in the description of $\mathcal{P}_i$.

Additionally, define the following cost function

$$J(U_0^{T-1}, x(0)) := \|Px(T)\|_p + \sum_{k=0}^{T-1} \|Qx(k)\|_p + \|Ru(k)\|_p \tag{2}$$

and consider the *finite time constrained optimal control problem* (CFTOC)

$$J^*(x(0)) := \min_{U_0^{T-1}} J(U_0^{T-1}, x(0)), \tag{3}$$

$$\text{subj. to} \begin{cases} x(t+1) = f_{\text{PWA}}(x(t), u(t)), \\ x(T) \in \mathcal{X}^f \end{cases} \tag{4}$$

where the column vector $U_0^{T-1} := [u(0)', \ldots, u(T-1)']' \in \mathbb{R}^{mT}$ is the optimization vector, T is the time horizon, $\mathcal{X}^f$ is the terminal target region and $\|Qx\|_p$ with $p \in \{1, \infty\}$ in (2) denotes the corresponding standard vector 1- or ∞-norm. Additionally, we assume that R, Q, and P are of full column rank.

We summarize the main result of the solution to the CFTOC problem (1)–(4) which is proved in [2].

Theorem 2.1 (Solution to the CFTOC). The solution to the optimal control problem (1)–(4) with $p \in \{1, \infty\}$ is a piecewise affine state feedback control law of the form

$$u^*(t) = f_t(x(t)) = F_i^t x(t) + G_i^t \qquad \text{if} \quad x(t) \in \mathcal{R}_i^t \tag{5}$$

where $\mathcal{R}_i^t$, $i = 1, \ldots, N^t$, is a polyhedral partition of the set $\mathcal{X}^t$ of feasible states $x(t)$ at time $t = 0, \ldots, T-1$. □

3 Computation of the CFTOC Solution via mp-MILP

One way of solving the constrained finite time optimal control problem (1)–(4) is by reformulating the PWA system into a set of inequalities with integer variables as switches between the different dynamics of the hybrid system. An appropriate modeling framework for such a class of systems are *mixed logical dynamical* (MLD) systems [8] where the switching behavior as well as the constraints of the system are modeled with inequality conditions. In [9] the authors show the equivalence between PWA and MLD systems.

Using an MLD representation the CFTOC problem (1)–(4) can be stated in the form

$$J^*(x(0)) := \min_{U_0^{T-1}} \|Px(T)\|_p + \sum_{k=0}^{T-1} \|Qx(k)\|_p + \|Ru(k)\|_p, \tag{6}$$

$$\text{subj. to} \begin{cases} x(t+1) = Ax(t) + B_u u(t) + B_\delta \delta(t) + B_z z(t), \\ E_\delta \delta(t) + E_z z(t) \le E_u u(t) + E_x x(t) + E, \\ x(T) \in \mathcal{X}^f \end{cases} \tag{7}$$

where $\delta \in \{0,1\}^{m_\delta}$ is the vector of integer variables and $z \in \mathbb{R}^{m_z}$ represents the vector of auxiliary variables, cf. [8].

By using an upper bound ε_t^x for each of the components, e.g. $\|Qx(t)\|_p \le \varepsilon_t^x$, of the cost function (6) and

$$x(t+1) = A^t x(0) + \sum_{j=0}^{t-1} A^j \big\{ B_u u(t-1-j) + B_\delta \delta(t-1-j) + B_z z(t-1-j) \big\} \tag{8}$$

the CFTOC problem can be rewritten as a mixed-integer linear program (MILP)

$$\min_{\varepsilon} \quad c'\varepsilon, \tag{9}$$

$$\text{subj. to} \quad G\varepsilon \le W + Sx(0) \tag{10}$$

where G, W, and S are matrices of suitable dimension, $c = [0_{1\times(m+m_\delta+m_z)T} \; 1 \ldots 1]'$, and the optimization variable is of the form $\varepsilon := [u(0)' \ldots u(T-1)' \, \delta(0)' \ldots \delta(T-1)' \, z(0)' \ldots z(T-1)' \, \varepsilon_0^{x\prime} \ldots \varepsilon_T^{x\prime} \, \varepsilon_0^{u\prime} \ldots \varepsilon_{T-1}^{u\prime}]'$. Note that $x(0)$ can be considered as parameter of the mp-MILP. The matrices G, W, and S contain the whole information on the state and input constraints, the weighting matrices P, Q, and R, as well as the update equation (8) for the whole time horizon T. Note that for the construction of S it is necessary to compute $A^t, t = 0, \ldots, T$.

For a given initial state $x(0)$ the MILP (9)–(10) can be solved in order to obtain the optimizer $\varepsilon^*(x(0))$ which in turn provides the optimal control sequence U_0^{T-1}. For exploring the whole feasible state space a multi-parametric MILP has to be solved. Dua and Pistikopoulos [7] proposed to split the original mp-MILP problem into two subproblems: an mp-LP and an MILP. The solution is found by recursion between these two subproblems by first fixing the integer variable and solving an mp-LP for this situation in order to explore and partition the feasible space. The intermediate solution gives an upper bound on the optimal cost. Then a new integer variable is fixed, an mp-LP is solved, and in case of overlapping polyhedral regions of the state space the resulting cost is compared with the previous one. In order to limit the exploration of the state space, which grows exponentially with the time horizon and the number of possible switching sequences, a branch and bound technique is applied.

4 Computation of the CFTOC Solution via an efficient Dynamic Program

Here we show that the considered constrained finite time optimal control problem (1)–(4) can be solved in a computationally more efficient way than with the mp-MILP method described in Section 3. In [1] Borrelli *et al.* solved such a problem for a quadratic performance index with an efficient dynamic programming approach.

For the 1- or ∞-norm case considered here the CFTOC problem can be formulated in a similar way as in [1]. The equivalent dynamic program is of the following form

$$J_j^*(x(j)) := \min_{u(j)} \|Qx(j)\|_p + \|Ru(j)\|_p + J_{j+1}^*(f_{\text{PWA}}(x(j), u(j))), \tag{11}$$

$$\text{subj. to} \quad f_{\text{PWA}}(x(j), u(j)) \in \mathcal{X}^{j+1} \tag{12}$$

for $j = T-1, \ldots, 0$, with boundary conditions

$$\mathcal{X}^T = \mathcal{X}^f, \text{ and} \tag{13}$$

$$J_T^*(x(T)) = \|Px(T)\|_p \tag{14}$$

where

$$\mathcal{X}^j = \big\{ x \in \mathbb{R}^n \mid \exists u, \; f_{\text{PWA}}(x, u) \in \mathcal{X}^{j+1} \big\} \tag{15}$$

is the set of all initial states for which the problem (11)–(12) is feasible.

In order to give a more detailed description of how the algorithm works some preliminary results have to be presented.

Basic Parametric Programming

All following results were stated and proved in [2] for the mp-QP case but also hold for mp-LP.

Consider the multi-parametric program

$$J^*(x) := \min_u \; l(x,u) + q(f^+(x,u)), \tag{16}$$

$$\text{subj. to } f^+(x,u) \in \mathcal{S} \tag{17}$$

where $\mathcal{S} \subseteq \mathbb{R}^n$, $f^+ : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^n$, $q : \mathcal{S} \mapsto \mathbb{R}$, and $l : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}$ are piecewise affine functions of x and u. Denote with $\mathcal{X}$ the set of variables x for which the parametric program (16)–(17) is feasible. Note that $q(\bullet)$ corresponds to the cost-to-go function $J^*_{j+1}(\bullet)$, cf. (11)–(12). First we define:

Definition 4.1 (PWA function with multiplicity). A function $q : \Theta \mapsto \mathbb{R}$, where $\Theta \subseteq \mathbb{R}^s$, is a multiple PWA function of order $d \in \mathbb{N}^+$ if $q(\theta) = \min \left\{ q^1(\theta) := l^1\theta + c^1, \ldots, q^d(\theta) := l^d\theta + c^d \right\}$ and Θ is a convex polyhedron. □

Here we summerize the main result needed for solving the CFTOC problem via dynamic programming. The reader is referred to the references [1, 2, 10] for details.

Result 4.2.

(a) *one to one problem*: f^+ is a linear function, q is a piecewise affine function, and $\mathcal{S}$ is a convex polyhedron. A one to one problem is solved with one mp-LP.

(b) *one to one problem of multiplicity d*: f^+ is a linear function, q is a multiple piecewise affine function of multiplicity d. A one to one problem of multiplicity d is solved by solving d mp-LPs.

(c) *one to many r problem*: f^+ is a linear function, q is a polyhedral piecewise affine function defined over r polyhedral regions. A one to many r problem is solved with r mp-LPs.

(d) *one to many r problem of multiplicity d*: f^+ is a linear function and q is a multiple polyhedral piecewise affine function of multiplicity d defined over r polyhedral regions. A one to many r problem of multiplicity d is solved with rd mp-LPs.

If the function f^+ is polyhedral piecewise affine defined over s regions then we have a *many s to X problem* where X can belong to any of the combinations listed above, i.e. we have a *many s to many r problem of multiplicity d* if f^+ is polyhedral piecewise affine defined over s regions and q is a multiple polyhedral piecewise affine function of multiplicity d, defined over r polyhedral regions.

(e) A *many s to one problem* can be decomposed into s one to one problem.

horizon T	Dynamic Programming CPU-time [sec]	Dynamic Programming # regions	mp-MILP CPU-time [sec]	mp-MILP # regions
1	0.3	10	1.4	10
2	2.1	16	12.3	17
3	6.1	26	36.2	59
4	20.8	52	125.8	220
5	58.3	90	317.5	441
6	155.1	152	912.6	819
7	358.8	218	2192.3	1441
8	662.0	262	5320.1	2257
9	973.6	268	$\star$	$\star$
10	1250.0	258	$\star$	$\star$
11	1515.0	252	$\star$	$\star$

Tab. 1: Comparison of the CPU-time in seconds and the number of regions for Example (20). $\star$ denotes that the computation for the particular problem did not converge.

(f) A *many s to many r problem* can be decomposed into s one to many r problems.

(g) A *many s to many r problem of multiplicity* d can be decomposed into s one to many r problem of multiplicity d. □

The Dynamic Programming Strategy

The dynamic programming problem (11)–(14) can be solved by using a multi-parametric linear program solver going backwards in time starting from the target region $\mathcal{X}^f$.

Consider the first step of the dynamic program (11)–(14)

$$J^*_{T-1}(x(T-1)) := \min_{u(T-1)} \|Qx(T-1)\|_p + \|Ru(T-1)\|_p + J^*_T(f_{\text{PWA}}(x(T-1), u(T-1)), \tag{18}$$

$$\text{subj. to} \quad f_{\text{PWA}}(x(T-1), u(T-1)) \in \mathcal{X}^f. \tag{19}$$

The cost-to-go function $J^*_T(x)$ in (18) is piecewise affine, the terminal region $\mathcal{X}^f$ is a polyhedron and the constraints are piecewise affine. Problem (18)–(19) is a many s to one problem that can be solved with s mp-LPs, cf. Result 4.2(e).

At the second step $j = T-2$ the cost-to-go function $J^*_{T-1}(x)$ is polyhedral piecewise affine and the terminal set $\mathcal{X}^{T-1}$ is a union of N^r_{T-1} polyhedra where N^r_{T-1} is the number of polyhedra of J^*_{T-1}. Note that the constraints are still piecewise affine but $\mathcal{X}^{T-1}$ is not necessarily a convex set. Problem (11)–(14) becomes a many s to many N^r_{T-1} problem and from Result 4.2(f) can be solved by solving sN^r_{T-1} mp-LPs. From the third step $j = T-3$ to the last one $j = 0$ the cost-to-go function $J^*_j(x)$ is polyhedral piecewise affine with a certain multiplicity d_j, the terminal set $\mathcal{X}^j$ is again a union of N^r_j polyhedra and the constraints are piecewise affine. Therefore, problem (11)–(14) is a many s to many N^r_j problem with multiplicity d_j, that from Result 4.2(g) can be solved by solving $sN^r_j d_j$ mp-LPs. The resulting optimal solution will have the piecewise affine form (5).

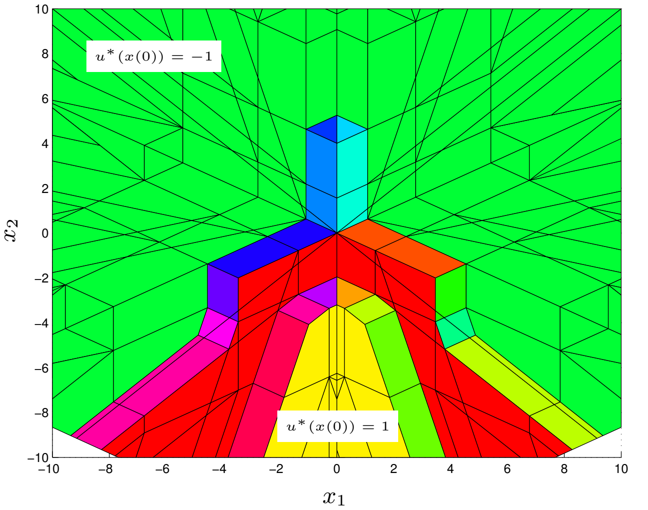

Fig. 1: State space partitioning of the finite time horizon solution for $T = 8$ derived with the dynamic programming algorithm. Same color corresponds to the same affine control law $u^*(x(0))$. There exist 19 different affine control laws in 262 polyhedral regions.

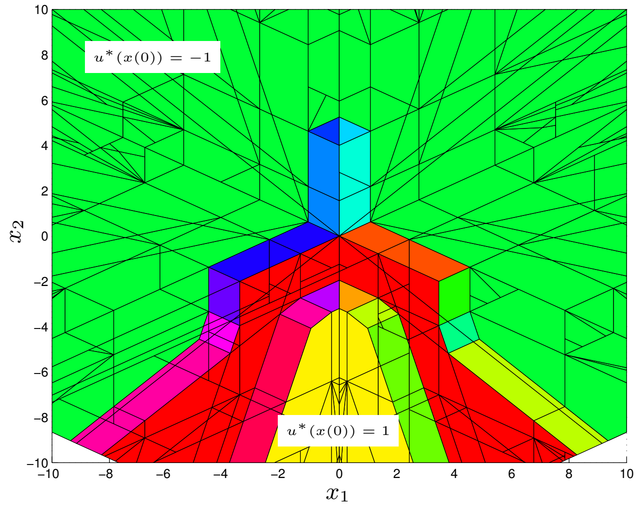

Fig. 2: State space partitioning of the finite time horizon solution for $T = 8$ derived with the mp-MILP algorithm. Same color corresponds to the same affine control law $u^*(x(0))$. There exist 19 different affine control laws in 2257 polyhedral regions.

5 Example

Consider the piecewise affine system [8]

$$\begin{cases} x(t+1) &= 0.8 \begin{bmatrix} \cos\alpha(t) & -\sin\alpha(t) \\ \sin\alpha(t) & \cos\alpha(t) \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t), \\ \alpha(t) &= \begin{cases} \frac{\pi}{3} & \text{if} \quad [1\ 0]x(t) \geq 0, \\ -\frac{\pi}{3} & \text{if} \quad [1\ 0]x(t) < 0, \end{cases} \\ x(t) &\in [-10, 10] \times [-10, 10], \\ u(t) &\in [-1, 1]. \end{cases} \tag{20}$$

The constrained finite time optimal control problem (1)–(3) is solved with $Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $R = 1$, $P = \left[\begin{smallmatrix} 0 & 0 \\ 0 & 0 \end{smallmatrix}\right]$, and $\mathcal{X}^f = [-10,\ 10] \times [-10,\ 10]$ for $p = \infty$.

In Table 1 we report the computational times and the total number of polyhedral regions of the solution to the aforementioned problem for various horizons T obtained with the dynamic programming algorithm and the mp-MILP algorithm. The computation was done on a Pentium 4, 2.2 GHz machine running MATLAB 6.1.

Figure 1 shows the state space partition of the finite time horizon solution for $T = 8$ computed with the dynamic programming algorithm. The same color corresponds to the same affine control law $u^*(x(0))$. There exist 19 different affine control laws in 262 polyhedral regions. Each polyhedral region corresponds to a different affine value function. Figure 2 shows the corresponding partition of the state space computed with the mp-MILP algorithm presented in Section 3. Here the 19 different affine control laws were found in 2257 polyhedral regions. Unnecessary slicing of the state space was produced by the recursive structure of the mp-MILP algorithm.

Figure 3 shows the state space partition for the infinite time horizon solution computed with the dynamic programming algorithm. A posteriori it can be shown with the dynamic programming procedure that the finite time solution for a horizon $T \geq 11 = T_\infty$ is in fact identical to the infinite time solution of the constrained optimal control problem. The same coloring scheme corresponds to the same affine control law. There exist 23 different affine control laws $u^*(x(0))$ in 252 polyhedral regions. Figure 4 reveals the corresponding value function for the state space partition. The same color corresponds to the same cost. The minimum cost is achieved at the origin. Figure 5 shows the state and control action evolution for an initial state of $x(0) = [-10\ 10]'$ for the infinite time solution obtained with the dynamic programming procedure.

6 General Comments

In this section some general remarks on what has been noticed as being important issues of the new technique compared with the mp-MILP approach will be given. It should be noted that both algorithms were implemented by the authors as MATLAB®-code. However the CPLEX® MILP-solver was used in the mp-MILP algorithm. No effort was made to optimize either of the algorithms.

Special Purpose

We believe that one of the main reasons for the efficiency of the dynamic programming algorithm compared to the mp-MILP algorithm, shortly described in Section 3, is that the dynamic programming approach solves the CFTOC problem in the extended (x, u)-space with inequality constraints, whereas the mp-MILP approach solves the same problem in the extended (x, u, z, δ)-space with equality constraints, which are numerically hard to handle and increase the size of the problem dramatically. This is because every equality constraint is translated into two inequality constraints for each time step for the whole time horizon.

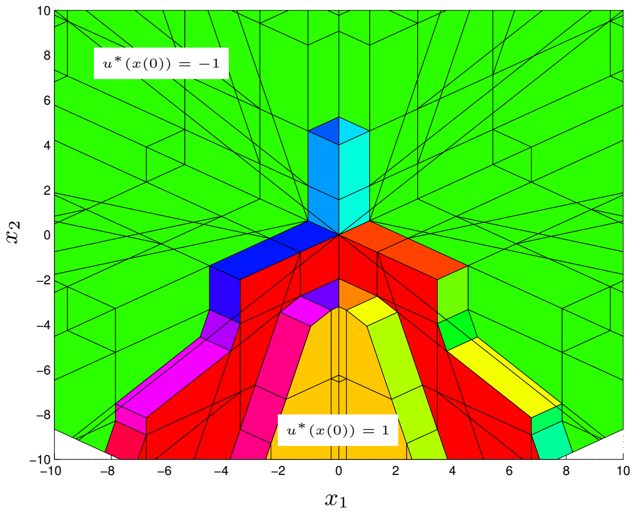

Fig. 3: State space partitioning of the infinite time horizon solution ($T = 11$) derived with the dynamic programming algorithm. Same color corresponds to the same affine control law $u^*(x(0))$.

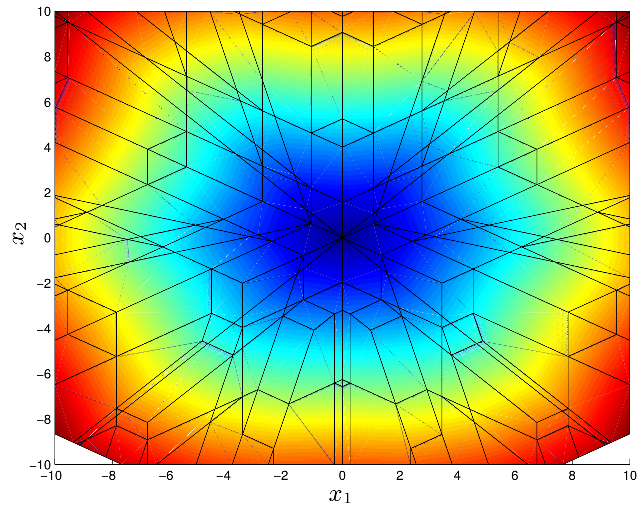

Fig. 4: State space partitioning of the infinite time horizon solution ($T = 11$) derived with the dynamic programming algorithm. Same color corresponds to the same cost value.

Another reason is that the dynamic programming approach is tailored to the considered class of optimal control problems. Contrary to that, the mp-MILP method solves general MILP problems and is therefore suitable for a larger class of problems such as for example optimal control problems with binary states or a large number of binary inputs as well as, for example, problems originating from economics or logistics.

Complexity of the Solution

It is observed by extensive simulations, cf. for example Table 1, that the dynamic programming approach is by a factor 5 to 10 times faster than mp-MILP algorithm (depending on the time horizon). Additionally the mp-MILP algorithm fails to find a solution for bigger time horizons because of the bigger memory demand.

Due to the possible degeneracy of linear programs it is impossible to provide a unique solution in all cases. Therefore it is very difficult to give a minimal representation for all possible occurring problems. However the obtained number of polyhedral regions in our approach was always smaller than in the mp-MILP approach. Because of the recursive structure of the mp-MILP method unnecessary slicings of the state space are introduced.

Ill-Conditioning

One of the reasons why the dynamic programming method is numerically more reliable is that for large time horizons T a computation of the possibly ill-conditioned matrix A^T is not needed while it is inherent in the MLD structure for the mp-MILP method, cf. Section 3. Furthermore, MLD systems do not exploit explicitly the structure of the control problem and for large time horizons MLD systems might become ill-defined because strict inequalities have to be modeled with slack variables that vanish to 0 as the time horizon increases.

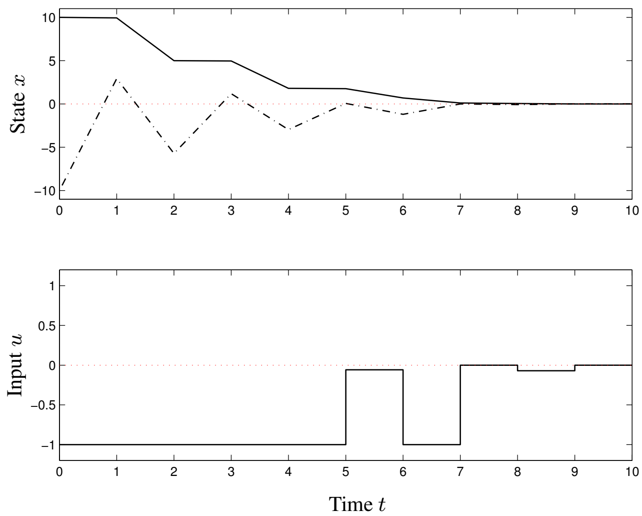

Fig. 5: State and control action evolution of the infinite time horizon solution derived with the dynamic programming algorithm. Initial state $x(0) = [-10\ 10]'$.

Infinite Time Horizon Solution

An important advantage of the dynamic programming approach is that after every step, starting from $t = T - 1$ to $t = 0$, the data of all the intermediate optimal control steps, the polyhedral partition of the state space, and the piecewise affine cost laws are available. This makes it possible to detect if the solution for a specific time horizon is identical to the *infinite time horizon solution* ($T \to \infty$), i.e. if for $T = T_\infty$ the whole feasible polyhedral state space partition and the cost as a function of the initial state $x(0)$ is identical to the polyhedral partition and the cost for $T \geq T_\infty$, respectively.

In some parts of the state space it is likely to happen that in two successive steps of the dynamic programming algorithm identical regions –in terms of the regions' dimensions and the associated value function– are generated. Such a case is depicted in

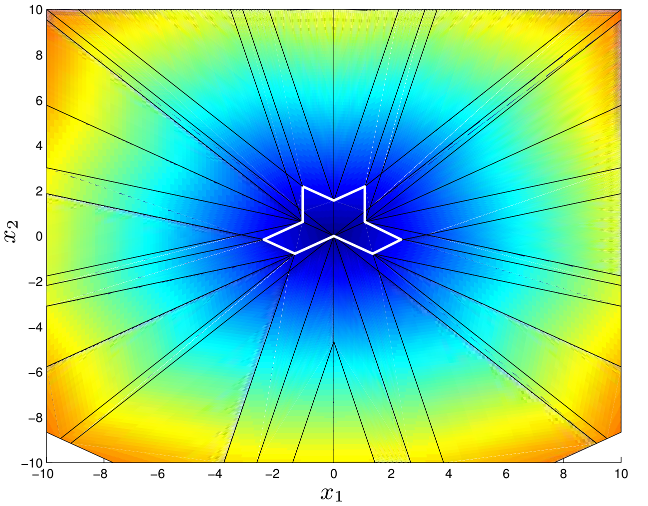

Fig. 6: State space partitioning of the finite time horizon solution for $T = 4$ derived with the dynamic programming algorithm. Same color corresponds to the same cost value. The white marked region is identical to the infinite time solution.

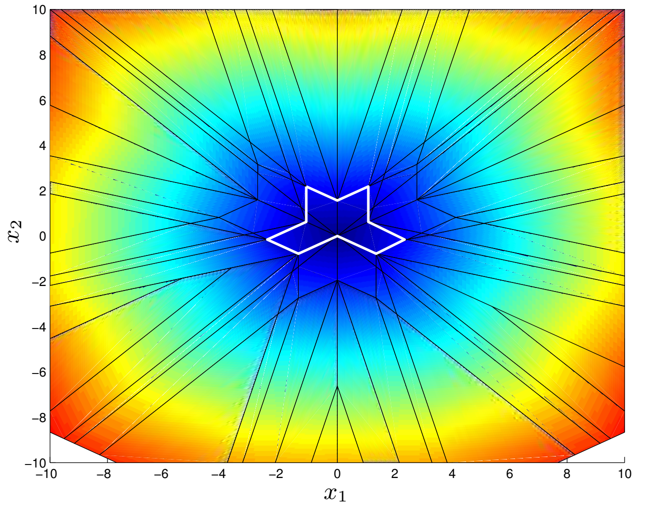

Fig. 7: State space partitioning of the finite time horizon solution for $T = 5$ derived with the dynamic programming algorithm. Same color corresponds to the same cost value. The white marked region is identical to the infinite time solution.

Figure 6 ($T = 4$) and Figure 7 ($T = 5$) for Example (20) where the white encircled regions are identical. This intermediate information could be used to speed up the proposed algorithm. (We would like to emphasize that the CPU-time reported in Table 1 is obtained with the dynamic programming algorithm which does not use such an intermediate information.) However, only when the algorithm converges in the whole feasible state space we can claim that the infinite time solution is obtained in any part of the state space. As a consequence it would be wrong to deduce that the infinite time solution was obtained in parts of the state space for some $T < T_\infty$. Such a claim can only be made a posteriori, i.e. after computing the solution to the CFTOC problem with $T \geq T_\infty$.

In [11] we propose a modification of the algorithm that constructs the infinite time solution in an efficient way by limiting the exploration of the state space in intermediate steps of the dynamic programming.

Other Advantages

Other advantages of the new algorithm are that only an mp-LP and not an mp-MILP solver is needed. Recursive calls like in the mp-MILP algorithm, which cause computational instability problems for large time horizons, are avoided by construction.

Acknowledgment

We wish to thank Francesco Borrelli for his continuous help and the encouraging suggestions.

References

[1] F. Borrelli, M. Baotić, A. Bemporad, and M. Morari. An Efficient Algorithm for Computing the State Feedback Solution to Optimal Control of Discrete Time Hybrid Systems. In *Proc. on the American Control Conference*, pages 4717–4722, Denver, Colorado, USA, June 2003.

[2] F. Borrelli. *Discrete Time Constrained Optimal Control*. Dr. Sc. Techn. thesis, Swiss Federal Institute of Technology (ETH), Zürich, Switzerland, May 2002.

[3] M.S. Branicky and G. Zhang. Solving hybrid control problems: Level sets and behavioral programming. In *Proc. on the American Control Conference*, Chicago, Illinois USA, June 2000.

[4] E.D. Sontag. Nonlinear regulation: The piecewise linear approach. *IEEE Trans. on Automatic Control*, 26(2):346–358, April 1981.

[5] A. Bemporad, F. Borrelli, and M. Morari. Optimal controllers for hybrid systems: Stability and piecewise linear explicit form. In *Proc. 39th IEEE Conf. on Decision and Control*, Sydney, Australia, December 2000.

[6] J. Lygeros, C. Tomlin, and S. Sastry. Controllers for reachability specifications for hybrid systems. *Automatica*, 35(3):349–370, 1999.

[7] V. Dua and E.N. Pistikopoulos. An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of Operations Research*, 99:123–139, 2000.

[8] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and contraints. *Automatica*, 35(3):407–427, March 1999.

[9] W.P.M.H. Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, 37(7):1085–1091, July 2001.

[10] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38(1):3–20, 2002.

[11] M. Baotić, F.J. Christophersen, and M. Morari. Infinite Time Optimal Control of Hybrid Systems with a Linear Performance Index. Technical Report AUT03-04, Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), March 2003. Available from: `http://control.ee.ethz.ch`.